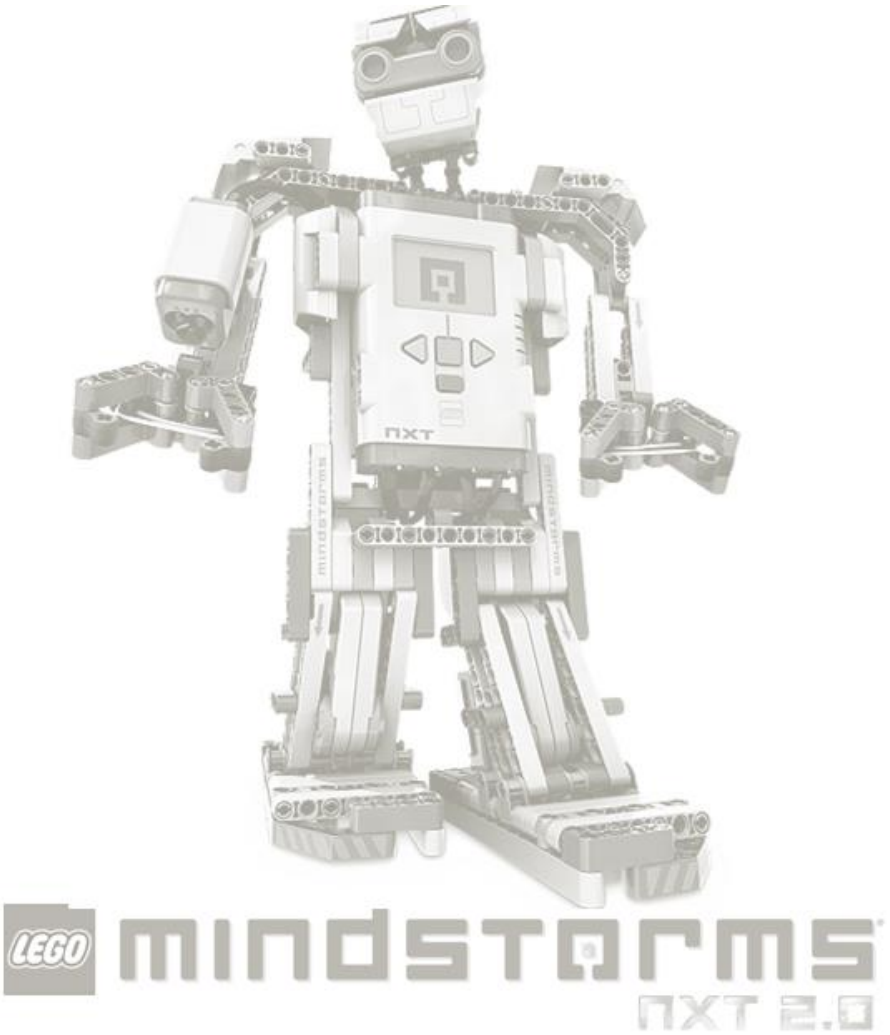
Universidade Federal Fluminense
TEE - Departamento de Engenharia Elétrica
Programa de Educação Tutorial da Engenharia Elétrica - PET-Elétrica

Apostila de Robótica Educacional

Versão 1.0



Autores: Elson Antonio Nunes Júnior
Hernani Justo da Silva Júnior
Marcus Vinicius Bouças
Rodrigo De Oliveira Siqueira

Niterói – Rio de Janeiro

2013

Sumário

1 - INTRODUÇÃO 4
2 - O QUE É PROGRAMAÇÃO? 4
3 - O KIT LEGO MINDSTORMS NXT 6
3.1 – NXT 7
3.2 - SERVO-MOTORES 8
3.3 - SENSORES 9
4 - VISÃO GERAL DO SOFTWARE 11
4.1- PALETA DE COMANDOS 12
4.2 - CONTROLADOR NXT 14
5 - BLOCOS DE PROGRAMAÇÃO 14
5.1 - MOVE BLOCK 14
5.2 - DISPLAY BLOCK 16
5.3 - TOUCH SENSOR BLOCK: 18
5.4 - LIGHT SENSOR BLOCK: 19
5.5 - ULTRASONIC SENSOR BLOCK: 21
5.6 - LOOP BLOCK: 22
5.7- SWITCH BLOCK 25
5.8 - WAIT BLOCK: 26
5.9 - VARIABLE BLOCK: 26
5.10 - MATH BLOCK: 27
5.11- Number To Text BLOCK 28
5.11 - USANDO FIOS DE DADOS (Data Wires) 28
6 - Experimento 1 - Práticas com o Sensor de Som NXT: 35
6.1 - Introdução: 35
6.2 - Sentidos e sensores: 35
6.3 - Experimento proposto: 37
6.4 - Entendendo o programa: 37
6.5 - Atividades: 38
7 - Experimento 2 – Lançador NXT: 39
7.1 – Introdução: 39
7.2 – Estudo do movimento dos corpos: 39
7.3 - Experimento Proposto: 41
7.4 - Entendendo o programa: 42

7.5 - Atividades:.....42
8 - Experimento 3 - Usando os Sensores Eletrônicos NXT.....43
8.1 - Introdução: 43
8.2 – O Sensor Ultrassônico 43
8.3 - Experimento proposto: 44
8.4 - Entendendo o programa 45
9 - Experimento 4 - Fita Métrica Eletrônica NXT: 46
9.1 - Introdução: 46
9.2 - Curiosidades sobre medidas e unidades: 46
9.3 - Experimento proposto: 49
9.4 - Entendendo o programa: 49
9.5 - Atividades:.....50
10 - Desafios.....51

1 - INTRODUÇÃO

Organizadas pelo Programa de Educação Tutorial de Engenharia Elétrica da Universidade Federal Fluminense (PET-Elétrica UFF), em parceria com o Departamento de Engenharia Elétrica (TEE) e a Escola de Engenharia da UFF, as aulas de **Robótica Educacional Como Ferramenta de Aprendizado para o Ensino Médio** têm como objetivo trazer para estudantes de Ensino Médio atividades práticas que tracem paralelos entre disciplinas teóricas apresentadas no Ensino médio, como Física e Matemática.

Com o uso dos kits **LEGO MINDSTORMS NXT**, os alunos poderão criar experimentos onde será posto em prática a teoria das salas de aulas, aplicada a projetos. Além disso, os alunos terão a chance de se aproximar de certos assuntos, que à primeira vista podem parecer distantes e complicados, mas que com o incentivo certo, se tornam enormes fontes de aprendizado e diversão. Estamos falando de **Robótica e Programação**.

Nessa apostila estão contidas: informações sobre os componentes do kit LEGO MINDSTORMS NXT 2.0 e suas funções; um tutorial básico sobre a *programação* envolvida na criação dos modelos, com exercícios, descrição de quatro experimentos com o kit Lego, que serão realizados para o estudante perceber as aplicações de conhecimentos de física e matemática, vistos na prática; e, ao final do curso, um Desafio Final a ser executado pelos alunos, para competirem no I Campeonato de Robôs da UFF, a se realizar em Outubro.

2 - O QUE É PROGRAMAÇÃO?

Programas são, basicamente, instruções que um computador recebe e executa conforme lhe foi dito. Algo parecido com uma receita de bolo: você quando quer preparar um bolo, deve seguir uma lista de comandos em uma ordem específica, para que consiga um resultado esperado; você não pode pular etapas, como por exemplo, acrescentar cobertura no bolo antes de levá-lo ao forno; não pode usar 5 ou 6 ovos para fazer a massa do bolo; resumindo, não pode ir além das instruções, se você quiser que o resultado final seja o bolo. Assim como a receita de bolo é para você, o programa será para o robô.

O grande desafio, porém, é que o robô não possui um cérebro como um humano, capaz de interpretar informações; portanto, as instruções que você der para ele têm que ser **específicas**.

Como pequeno exemplo: Você quer construir um carro robô que desvie de obstáculos a frente dele sozinho. Para um ser humano dirigindo este carro, os comandos que você daria para ele seriam exatamente esses:

Ex.:

"Dirija o carro até que, se encontrar um objeto desvie, e continue a dirigir."

Mas, para um carro-robô, os comandos precisam ser bem mais específicos:

"Ligue os motores.
Siga em frente até que:
Se encontrar um objeto a frente:
Pare os motores;
Gire a direção duas vezes para esquerda;
Siga em frente.
Se não encontrar um objeto a frente:
Siga em frente."

A programação no software oficial da linha LEGO MINDSTORMS, a linguagem **NXT-G,** é do tipo *"programação em blocos"*, isto é, possui um ambiente interativo de arrastar blocos de comandos para fazer a cadeia de programação. Esta apostila cobrirá os comandos básicos existentes no programa, o necessário para se realizar os experimentos contidos no final da apostila.

3 - O KIT LEGO MINDSTORMS NXT

O kit de Robótica Educacional com o qual aprenderemos é o kit *9797 LEGO MINDSTORMS Education NXT Base Set*. Diferente dos blocos de montar Lego mais comuns, o kit consiste, além de suas peças, de três servo-motores que fazem o robô se mover, vários sensores que o robô usa para interagir com o ambiente ao seu redor, e do NXT Intelligent Brick (ou somente NXT, visto na fotografia abaixo), considerado o cérebro do robô.

No total, o Kit Lego Mindstorms NXT 9797 é composto de 437 peças, sendo:

- 01 Bloco lógico programável LEGO NXT;
- 01 Sensor de Luminosidade; 01 Sensor de Som;
- 02 Sensores de Toque; 01 Sensor de Ultrassom;
- 03 servomotores com encoder´s;
- 07 Cabos conectores (Padrão RJ-12 polarizado à direita);
- Peças LEGO, como: Blocos de tamanhos diversos;
- 02 lâmpadas (LED);
- Elementos estruturais, como vigas (em L, retas e angulares) e pranchas;
- Engrenagens (retas, cônicas e mistas, sistema “coroa-pinhão”);
- Correias, buchas e polias; Rodas, pneus e eixos de diversos tamanhos;
- Conectores, conectores com inversão e conectores em ângulos;
- Bateria recarregável de Íon-Lítio 1400mA;

3.1 – NXT

Figura 3.1 - NXT Intelligent Brick

O NXT é a central de processamento do robô, pois, a partir dele, é que ocorre o trânsito de informações, provenientes da programação que você inseriu nele, como a leitura de sinais do ambiente em que o robô está inserido. Ele possui quatro canais de entrada, onde os sensores são inseridos, três canais de saída, onde os servo-motores são inseridos, e uma entrada USB, por onde se conecta a um computador para inserir o arquivo onde está escrita a programação.

Também possui quatro botões de controle do robô (que também podem ser programados para realizar ações se apertados), um sistema de alto falante do lado esquerdo, para reproduzir sons, e uma tela de LCD que pode ser programada para demonstrar a atividade dentro do NXT. Associados ao NXT estão diversos **blocos de programação**, como:

- **NXT Buttons Block,**
- **Display Block,** entre outros.

Os blocos de programação serão explicados mais adiante no capítulo 5.

3.2 - SERVO-MOTORES

Os servo-motores são pequenos motores que realizam todas as tarefas mecânicas do seu robô. Cada kit vem com três servo-motores, que sempre são conectados aos canais de saída A, B e C do NXT. A eles estão associados blocos de programação tais como:

Figura 3.2 - Servo-motor

- **Move Block;**
- **Motor Block**;
- **Calibrate Block;**
- **Rotation Sensor Block;**
- **Reset Motor Block.**

A figura abaixo representa o funcionamento do servo-motor por dentro, composto de várias engrenagens de transmissão:

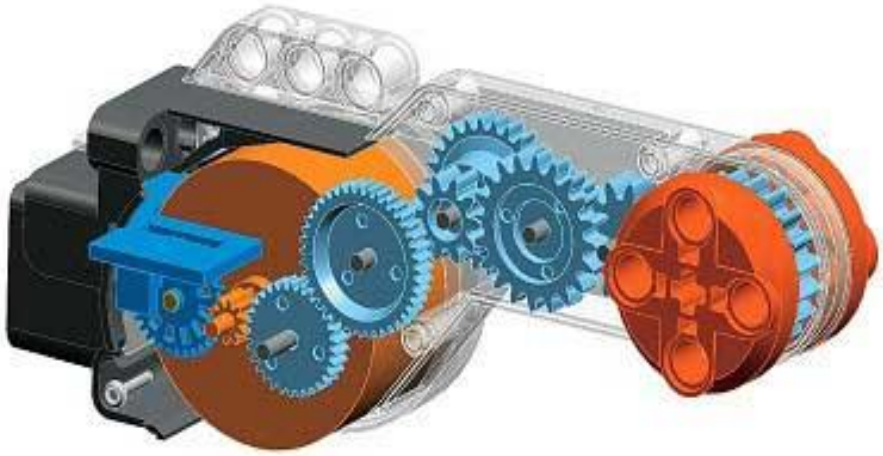


Figura 3.3 - Seção lateral do servo-motor NXT, mostrando o sistema de engrenagens de transmissão de movimento.

3.3 - SENSORES

Sensores são dispositivos eletrônicos que também recebem comandos da programação e realizam as mais variadas tarefas. Eles também são úteis para recolher informações do local onde o robô está. Essas informações são convertidas em dados, interpretados pelo NXT Intelligent Brick, que reage de acordo com a programação descrita no código. São conectados às quatro portas de entrada 1, 2, 3 e 4. No kit LEGO coberto por essa apostila, os sensores são: *Light Sensor (Sensor de Luminosidade)*, *2 Touch Sensors (Sensores de toque)*, um *Ultrasonic Sensor (Sensor Ultrassônico)*, e um *Sensor de Som (Sound Sensor)*.

3.3.1- SENSOR DE LUMINOSIDADE

O sensor de luminosidade é capaz de captar diversos níveis de luminosidade, seja de luz ambiente, ou luz refletida em alguma superfície. Ele também pode emitir luz, através de um pequeno LED na ponta do sensor. O Sensor de Cor pode detectar seis cores: Vermelho, azul, verde, amarelo, branco e preto, enquanto a luz emitida pelo LED pode ser vermelha, azul ou verde.

Figura 3.4 - Sensor de Luminosidade

A ele está associado o bloco **Light Sensor Block**.

3.3.2- SENSOR DE TOQUE

Os sensores de toque servem para criar a sensação de tato para o seu robô, para que ele possa perceber e reagir ao ambiente a sua volta através do toque. Os botões podem executar comandos quando apertados, pressionados, ou levemente apertados.

Figura 3.5 - Sensor de Toque

A eles estão associados o bloco **Touch Sensor Block**.

3.3.3- SENSOR ULTRASSÔNICO

Figura 3.6 - Sensor Ultrassônico

O sensor ultrassônico é capaz de emitir um leve sonar para rastrear o ambiente a sua frente, agindo como os olhos do robô. Com esse sensor, é possível determinar também a distância de objetos que estejam no caminho do robô.

A ele está associado o bloco **Ultrasonic Sensor Block**.

3.3.3- SENSOR DE SOM

O sensor de som funciona como um pequeno microfone, que detecta o valor sonoro em decibéis, e é capaz também de detectar diferentes timbres sonoros, agindo como os ouvidos do robô.

A ele está associado o bloco **Sound Sensor Block.**

Além desses sensores, também podem ser encontrados diversos outros que complementam o seu robô, deixando cada vez mais autônomo e complexo. Porém, são vendidos separadamente, como Sensores de Som (não necessários a esse kit, já que o próprio NXT possui um microfone interno), Temperatura, de Rotação, Acelerômetros, etc.

4 - VISÃO GERAL DO SOFTWARE

A imagem abaixo exibe a tela inicial de programação do LEGO MINDSTORMS NXT:

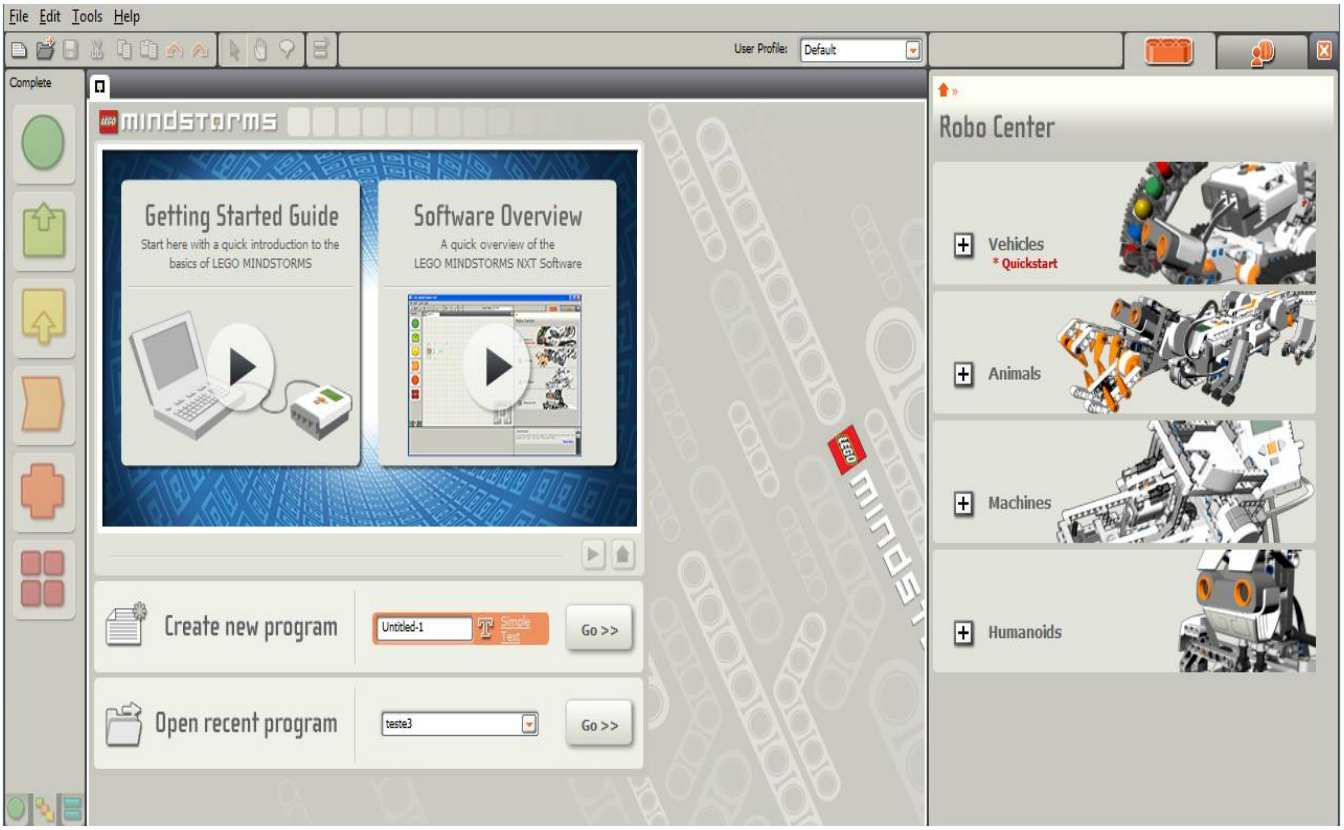


Figura 4.1 - Tela inicial do Programa LEGO MINDSTORMS NXT

A princípio, o programa abre a tela acima, dando as opções Create new program (*Criar novo programa)* ou Open recent program (*Abrir programa recente)*. À direita, aparece a guia Robo Center, que consiste de modelos de montagem de robôs já disponibilizados pela LEGO no software, que não serão abordados pela apostila.

Após abrir um novo arquivo de nome *"Untitled-1.rbt"*, nome inicial que você pode modificar, estamos prontos para começar a programar. A seguinte tela é aberta, mostrando a Paleta de Comandos e o Controlador do NXT:

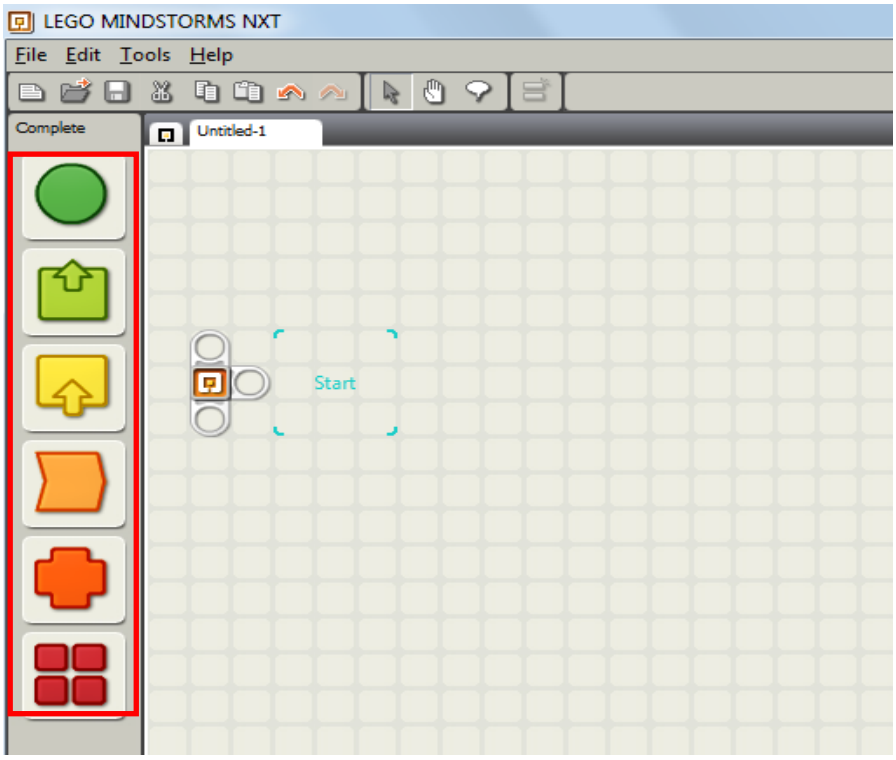


Figura 4.2 - Ambiente de programação do Programa LEGO MINDSTORMS NXT. Em destaque em vermelho, a Paleta de Comandos

4.1- PALETA DE COMANDOS

No canto esquerdo do ambiente de programação, encontram-se a Paleta de Comandos, onde estão os blocos que iremos usar para programar. Ela pode ser exibida de três formas: *Common, Complete, e Custom*. Usaremos apenas a exibição completa em nossa apostila.

Os blocos de programação se dividem na paleta completa em grupos de blocos, que são: **Common, Action, Sensor, Flow, Data, e Advanced**.

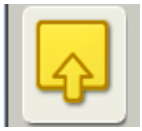
	Common: Nesse grupo encontram-se os blocos da paleta comum, que são os mais utilizados geralmente em um programa. Com eles podemos acionar os servo-motores, criar algumas repetições, usar o display do NXT para apresentar valores, dentre outras aplicações. Ex: **Move Block, Loop Block**, etc.
	Action: Blocos que realizam ações, bem parecido com os blocos do grupo acima. Destaque para o **Motor Block**, bloco usado para controlar com mais precisão apenas um servo-motor por vez, e o **Send Message Block**, usado para controlar remotamente o NXT via Bluetooth (não será utilizado nos exercícios).
	**Sensor:** Blocos usados pelos sensores, já mencionados anteriormente. Destaque para o **Timer Block**, não mencionado anteriormente, que registra o tempo que o programa está rodando, ou que um dos servo-motores está sendo executado, graças a três contadores internos de tempo, um para cada servo-motor.
	Flow: Grupo de blocos que contém **Estruturas de Controle**, que são blocos que guiam o seu robô a agir de acordo com condições que podem ser estabelecidas por nós, ou pelo local onde ele está. Essas estruturas tornam o programa, e consequente o robô, mais inteligente. Por exemplo, temos o **Loop Block**, **Wait Block, Stop Block**, e **Switch Block**
	Data: Blocos lógicos ou matemáticos, usados para fazer cálculos e comparações entre os valores obtidos pelos sensores e servo-motores, e, a partir daí, tomar ações. Como exemplo, temos: **Logic Block**, **Math Block**, e **Random Block**. O uso desses blocos torna seu programa muito mais preciso e dinâmico, como veremos nos exercícios.
	Advanced: Esse blocos são usados para funções mais complexas do Lego Mindstorms. Para o uso dessa apostila, serão utilizados somente o **Number to Text Block** e **File Access Block**, e seus usos serão devidamente explicados conforme necessário.

Tabela 4.1 – Paleta de comandos e seus grupos de blocos

4.2 - CONTROLADOR NXT

No canto inferior-direito, encontra-se o Controlador NXT, onde podemos fazer o download dos programas criados para o NXT, além de verificar características do NXT, como bateria restante, memória ocupada, etc.

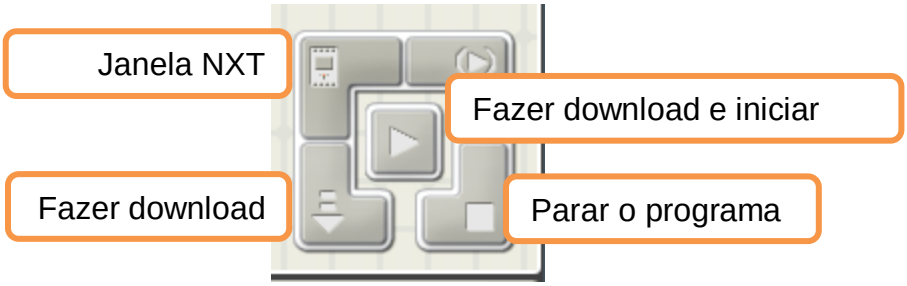


Figura 4.3 - Controlador NXT e descrição dos seus botões

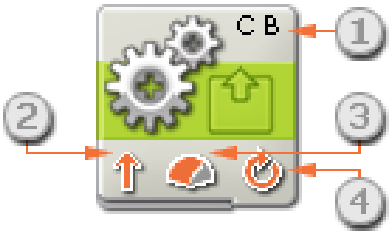
Agora que você já conhece o layout básico do programa, está na hora de começar a conhecer os Blocos de Programação.

5 - BLOCOS DE PROGRAMAÇÃO

A programação em blocos do LEGO MINDSTORMS NXT-G é a linguagem de programação principal para se programar o Kit LEGO MINDSTORMS. Cada bloco existente no programa possui certas configurações, que precisam ser observadas da mesma forma, para que o bloco faça exatamente o que se quer. Esta seção vai cobrir, além de uma breve descrição dos principais blocos, o conceito de **Fios de Dados** (*Data Wires*), necessário para trabalhar melhor com informações coletadas pelo programa ou pelos sensores.

5.1 - MOVE BLOCK

O Move Block serve para girar os servo-motores em sentido horário ou anti-horário, e dessa forma, fazer com que seu robô ande para frente ou para trás em uma linha reta, ou girar para poder fazer uma curva. Você pode também definir a distância percorrida pelo seu robô através da configuração de duração do movimento. Ao arrastar o Move Block para a linha de comandos, as seguintes **configurações padrão** que irão aparecer nele são:



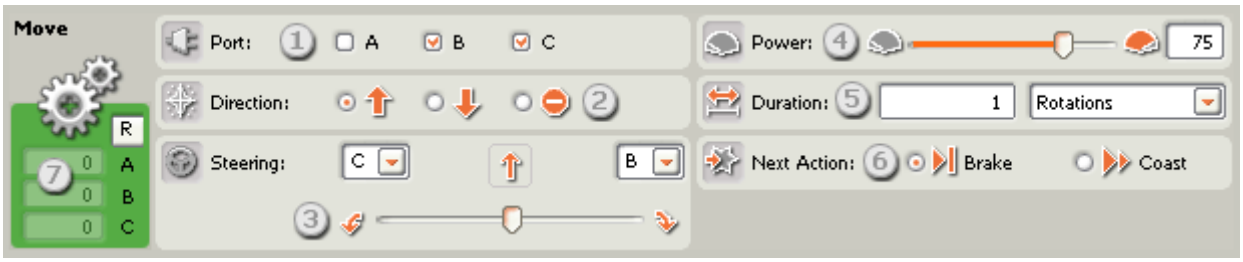
1. As letras no canto superior direito do bloco mostram quais portas de servo-motores (A, B ou C) o bloco irá controlar;

2. Esse ícone mostra em qual sentido seu servo-motor está girando.

3. Esse ícone mostra o nível de potência no motor. Quanto mais potência, maior a velocidade que seu robô irá andar, ou maior força ele terá para levantar coisas, ou subir uma rampa, por exemplo. Para fazer movimentos mais precisos, porém, é preciso menos potência.

4. Esse ícone mostra a duração do movimento. Pode ser configurado para contar o número de rotações do motor, graus de giro, segundos de movimento, ou infinitamente. Observe que os giros serão medidos de acordo com a engrenagem interna do servo-motor, portanto, precisam ser convertidos adequadamente para engrenagens ou rodas maiores.

5.1.1 - Configurando o Move Block



1. Seleciona os motores que você quer controlar. Caso você selecionar dois ou mais motores, eles se moverão em sincronia;

2. Escolhe o sentido de giro do motor (Horário ou anti-horário), ou informa para ele parar. Selecionando Pare (**Stop**) irá desligar os motores;

3. Caso você estiver usando dois ou mais motores para controlar um veículo, um de cada lado, a configuração **Steering** estará disponível. Ela serve para controlar a potência

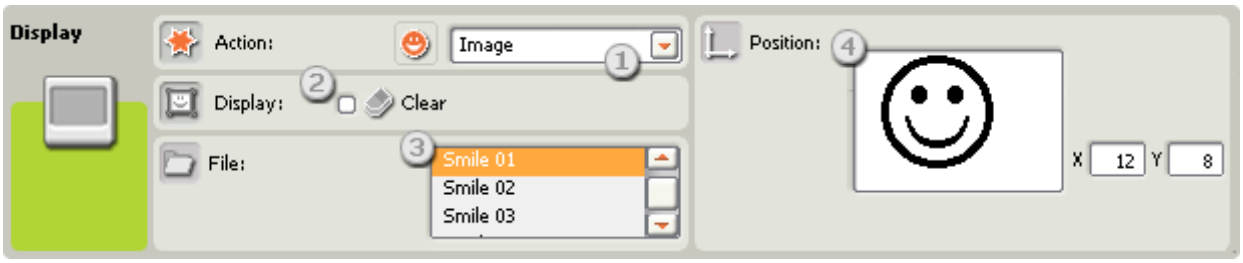
oferecida aos dois sensores de forma automática. Ao deslizar o botão, o motor da esquerda ou da direita receberá mais ou menos potência, o que é útil para fazer curvas;

4. Regula o nível de potência dada para os motores, de 0 a 100%;
5. Escolhe de qual forma será controlada a Duração do movimento do robô (por rotações do motor, por graus de giro, segundos decorridos ou de forma infinita). Controlando a duração do movimento, é possível controlar a distância percorrida pelo robô.
6. Escolhe a próxima ação do seu robô (após a duração do movimento ser respeitada). Os motores podem frear (*Brake*) ou estacionar (*Coast*). O comando **Brake** faz com que o motor pare com mais precisão. Ele também gasta mais energia, visto que realmente trava as engrenagens do seu motor. Já o comando **Coast** para os motores mais suavemente.
7. As três caixas de feedback fazem a contagem dos graus de giro ou rotações dadas pelo motores em tempo real, com o programa rodando, mas ainda ligado ao computador pelo cabo USB. Aperte o botão reset para zerar os valores exibidos.

5.2 - DISPLAY BLOCK

O Display Block serve para poder exibir uma imagem ou texto na tela de LCD do seu NXT. Ao inserir vários Blocos Display em sequência, você pode ir criando imagens mais complexas, ou adicionando mais texto na tela, conforme desejar. O uso da tela de LCD é importante para que você tenha uma espécie de controle do que está acontecendo dentro de seu programa, quando o robô estiver em funcionamento.

5.2.1 - Configurando o Display Block



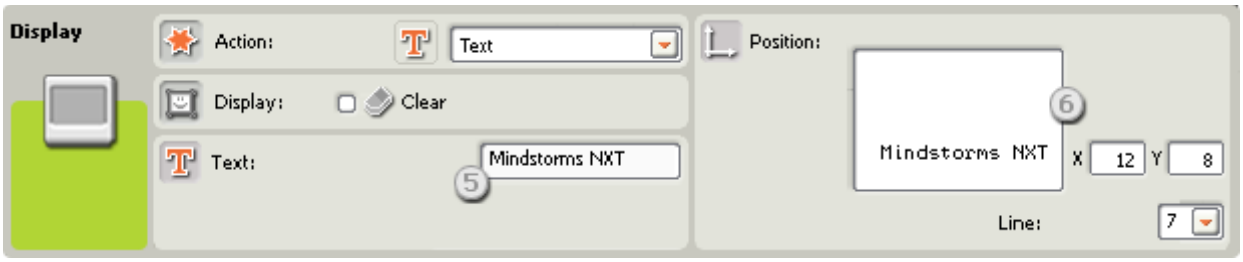
1. Nesse menu você pode escolher se vai exibir uma imagem, texto, ou um desenho próprio; ou escolher a opção **Reset**, para apagar o display.

2. A opção **Clear** serve para apagar o conteúdo da tela.

Para exibir uma imagem (como na imagem acima), você poderá configurar as seguintes opções:

3. Em **File** (*Arquivo*), você poderá escolher uma imagem do arquivo de imagens para exibir.

4. Em **Position** (*Posição*), você pode arrastar sua imagem exibida para qualquer canto da tela, ou inserir as coordenadas X e Y de onde exatamente você quer que a imagem seja impressa na tela.



Para exibir texto (como na imagem acima), você poderá configurar as seguintes opções:

5. Escrever o texto desejado na caixa **Text.**

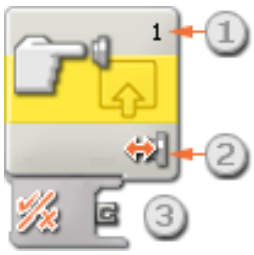
6. Em **Position** (*Posição*), você pode arrastar seu texto para ser exibido em qualquer canto da tela, ou inserir as coordenadas X e Y de onde exatamente você quer que o texto seja impresso na tela.

5.3 - TOUCH SENSOR BLOCK:

O **Touch Sensor Block** serve para monitorar as condições do Sensor de Toque em pontos específicos do seu programa. Ele pode ser acionado de três maneiras distintas (*encostado, pressionado e liberado)* e, quando acionado, envia essa informação para o NXT, que executa alguma ação com esse dado. Os sensores transportam essas informações coletadas através de **Fios de Dados** (*Data Wires*).

Os Fios de Dados são conectados no Input/Output de um bloco para outro, transportando a informação obtida pelo primeiro bloco. No caso do Touch Sensor Block, essa informação será um *valor lógico*, ou seja, se o botão foi apertado, o sensor envia um valor **verdadeiro** (*True*) para o NXT; senão, envia um valor **falso** (*False*). Em informática, os valores Tue/False são computados também como 0 e 1 pelo processador do NXT. Os Fios de Dados serão mais bem abordados na seção **5.11.**

Ao arrastar o Touch Sensor Block para a linha de comandos, as seguintes **configurações padrão** que irão aparecer nele são:

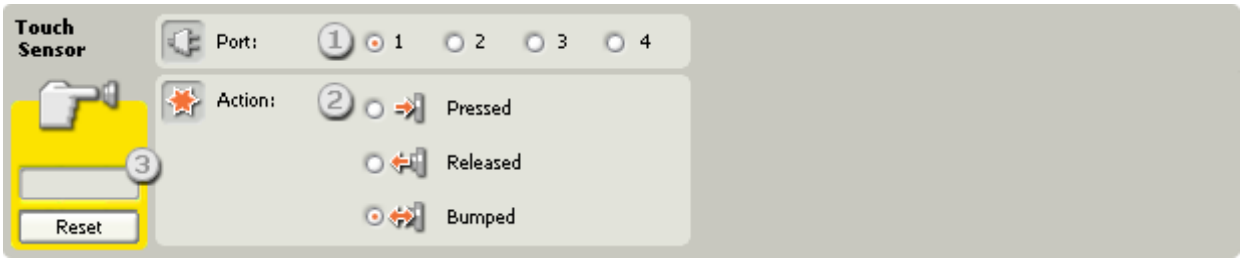


1. O número mostra qual a porta do NXT estão conectadas ao sensor de toque. A porta pode ser alterada na opção **Port**.

2. Esse ícone mostra qual ação enviará um valor lógico para o NXT. Se o sensor for acionado, envia um sinal **True**; Caso contrário, envia um sinal **False**.

3. O Input de valores lógicos do bloco abre automaticamente quando o bloco for arrasto para o programa. É possível expandí-lo, para mostrar mais opções, ou fechá-lo totalmente, clicando na linha sombreada cima do Input.

5.3.1 - Configurando o Touch Sensor Block



1. Seleciona a porta que o Sensor de Toque está conectado.

2. O Sensor de Toque pode ser acionado de três formas: Escolha **Bumped** (*encostado*) se você quer que o sensor seja acionado após ser pressionado e liberto rapidamente (menos de 0,5 segundos). Escolha **Pressed** (*pressionado*) se você quer que o sensor seja ativado no exato momento em que é pressionado. E por fim, escolha **Released** (*liberado*) se você quer ativar o valor lógico True quando o sensor for solto (caso contrário da opção Pressed).

3. A caixa de feedback permite que você teste seu sensor de toque, quando o robô ainda está conectado ao PC pelo cabo USB. Se você apertá-lo, dependendo da opção que marcou em (2), irá surgir o valor 0 ou 1 na caixa. Clique em **Reset** para limpar a caixa após o sensor ter sido apertado.

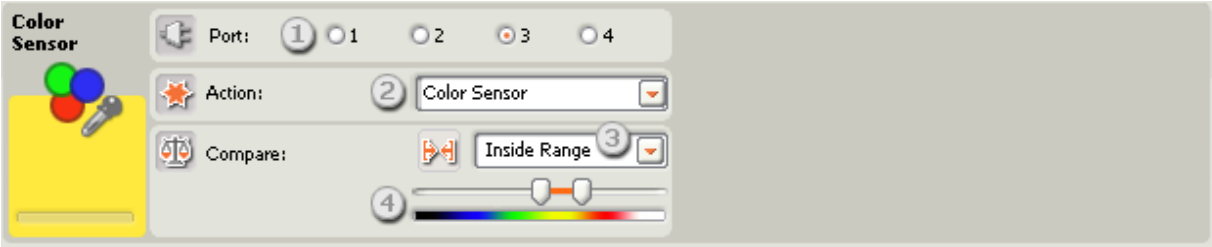
5.4 - LIGHT SENSOR BLOCK:

O sensor de cores do bloco possui dois modos: um para a detecção de cores diferentes e o outro para medir a intensidade da luz. Os modos podem ser usados dentro de condições impostas por loops ou mesmo por comando Wait, as informações podem ser tanto comparação de intensidade de luz quanto tonalidades.

1- Número da porta aonde o Sensor de cor esta ligado, este pode ser mudado pelas configurações do bloco.
2- Este ícone mostra o modo de sensor de cor.

3- O concentrador de dados do bloco irá abrir automaticamente, quando o bloco é colocado na área de trabalho. Você pode arrastar fios que ligam informações dos dados obtidos pelo bloco para outro.

5.4.1 Configurando no modo sensor de cor:



1- O número no canto superior direito do Color Sensor bloco mostra qual porta NXT estará controlando o bloco. Você pode mudar este número no painel de configuração, se você precisar.
2- Este ícone mostra o modo de sensor de luz e da cor selecionada na barra de função: vermelho, verde ou azul.
3- Este ícone indica a tonalidade da cor definida
4- Use os controles deslizantes direita e esquerda para definir a faixa de cores que irá definir o intervalo de cor: preto, azul, verde, amarelo, vermelho e branco.

5.4.2 - Configurando no modo Light Sensor

1- Use o controle deslizante para definir o valor desejado ou digite um valor diretamente na caixa de entrada. Selecione o botão à direita do controle deslizante com a opção maior (>) se você deseja comparar valores maiores do que o escolhido, escolha a opção menor (<) se deseja comparar valores menores que o escolhido.
2- Se você marcar a caixa de seleção "Light", o sensor de luz vai se transformar em sua própria fonte de luz e detectar essa luz, se ela é refletida de volta para ele. A luz padrão gerada será vermelha, mas você pode escolher verde ou azul dependendo da superfície. Se a luz estiver selecionada, a cor a ser acionada será mostrada como um

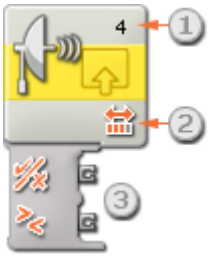
ícone no bloco. Deixe a caixa de seleção "Light" desmarcada para medir a luz ambiente.

3- A caixa de comentários mostra a quantidade atual de luz detectada (0-100). Você pode usá-la para experimentar diferentes valores.

5.5 - ULTRASONIC SENSOR BLOCK:

Este bloco pode detectar objetos a uma distância máxima de cerca de 100 polegadas (250 cm). Usando fios de dados, ele pode enviar a leitura de ultra-som e um sinal lógico (true / false) com base na leitura de ultra-som atual, verificando se esta se situa acima ou abaixo de um ponto escolhido.

O ponto inicial é o valor determinado de uma série de números em que uma alteração na condição ocorre. Por exemplo, você pode programar o seu robô para atacar quando um objeto se aproximar mais de 30 polegadas (76 cm) de alcance máximo do sensor ultra-sônico. O ponto de disparo seria 30.

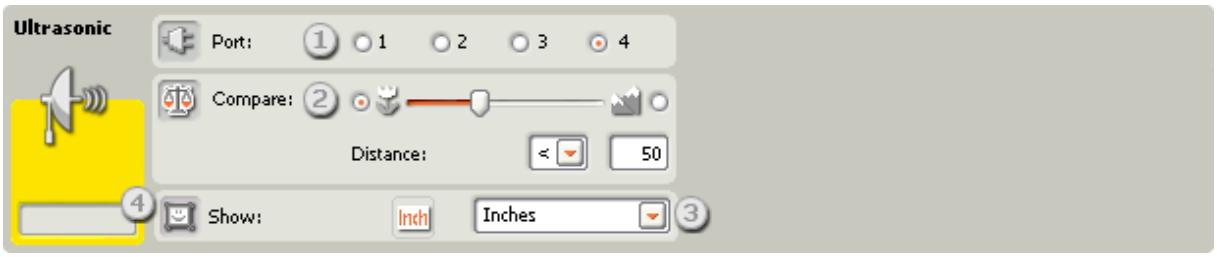


1- O número mostra qual das portas está conectado ao sensor ultra-sônico. Você pode mudar este número no painel de configuração, se você precisar.

2- Este ícone mostra se o sensor de ultra-som está definido para detectar objetos próximos ou distantes. As barras mais coloridas indicam distâncias maiores de detecção.

3- O concentrador de dados do bloco irá abrir automaticamente quando o bloco for colocado na área de trabalho. Você pode arrastar fios que ligam informações dos dados obtidos pelo bloco para outro.

5.5.1 - Configurando o bloco do sensor ultra-sônico:

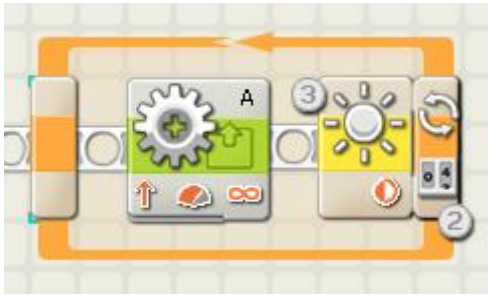


1- Define em qual porta você deseja por o sensor, por padrão este está na 4.
2- Nesta opção você pode usar a barra deslizante para escolher a distância de comparação que você deseja medir; na opção logo abaixo da barra você decide se irá procurar por valores maiores ou menores em relação ao escolhido.
3- Nesta opção você escolhe a unidade de comprimento, centímetros ou polegadas.
4- A caixa de diálogo te mostra qual a maior e a menor medida que pode ser mensurada, de acordo com cada unidade.

5.6 - LOOP BLOCK:

Utilize este bloco para repetir sequências do código. Defina uma condição que vai parar o loop; elas podem ser tempo decorrido, número de repetições, um sinal de lógica ou um sensor. Você também pode definir um loop que dure para sempre.

Algumas configurações:



1- Quando a propriedade de ação de um bloco de Loop é definido como "Forever", um símbolo de infinito (∞) é exibido na parte inferior do bloco.

2- Se a caixa de seleção "Show Contador" é selecionada no painel de configuração, um plug aparecerá que permitirá que você use o número de voltas concluídas como uma entrada em outro lugar no seu programa, você também pode usar a contagem atual para controlar o próprio loop.
3- Se você optar por um sensor para controlar o loop, a imagem do sensor escolhido irá aparecer na parte posterior. Por exemplo, se você escolher um sensor de luz para controlar o loop, um ícone do sensor de luz será exibido na parte ampliada do bloco. Além disso, todas as informações relevantes sobre a propriedade do controle escolhido serão exibidas na parte inferior do bloco.

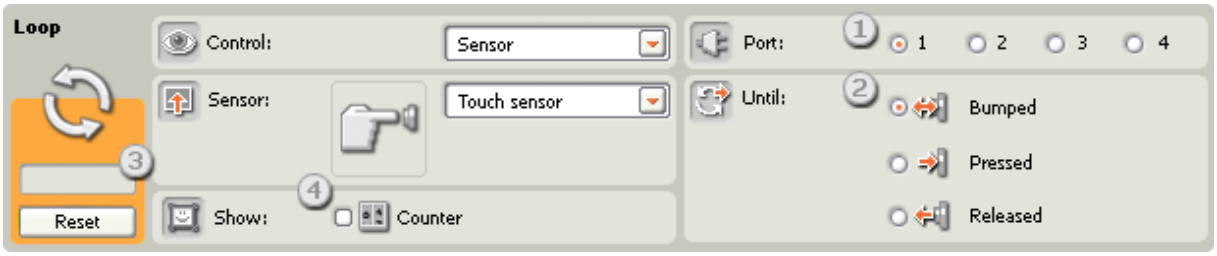
5.6.1 Configurações dos condicionais:

Forever

1- Marque a caixa "Show Contador", para usar o número de ciclos completos como entrada para outro bloco.

Os sensores possuem basicamente o mesmo princípio, mudando claro que cada um tem suas configurações. Apresentaremos a configuração do touch sensor e as demais serão vistas em práticas.

Touch Sensor

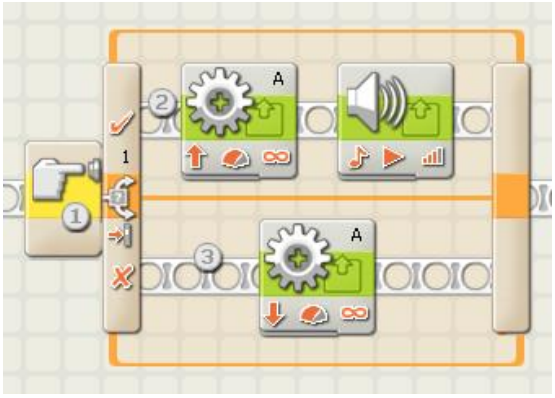


Se você escolher "sensor de toque", os blocos de programação dentro do loop serão repetidos até que um sensor de toque é colidido, pressionado ou liberado. Então, o ciclo vai terminar e o programa vai continuar.

1- Porta onde o sensor estará conectado.
2- Seleciona a condição na qual o loop estará restringido, relacionado ao sensor de toque neste caso.
3- A caixa de comentários deixará você testar o seu sensor de toque. Quando o sensor é ativado em seu robô, o número "1" será exibido aqui.
4- Se a caixa de seleção "Show Contador" é selecionada no painel de configuração, um plug-in aparecerá que permitirá que você use o número de voltas concluídas como uma entrada em outro lugar no seu programa.

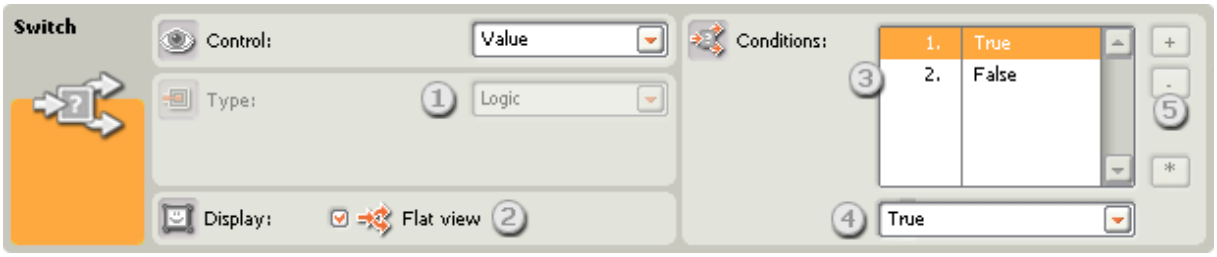
5.7- SWITCH BLOCK

O Switch Block é um bloco que tem a função de decisão entre duas sequências de códigos. Por exemplo, quando você configura o **TOUCH SENSOR,** o switch block pode executar uma série de blocos quando o botão for pressionado, ou outra sequência de blocos quando não pressionado. Seria como um bloco de decisão, que divergisse entre duas escolhas.



1. Esse ícone indicado pelo sensor mostra a condição imposta para escolher entre dois caminhos de linha de blocos.
2. Caso o botão seja pressionado, então será executado o **UPPER BLOCKS.**
3. Caso o botão não seja pressionado, será executado o **LOWER BLOCKS.**

Configurações:



1- Esta opção está relacionada com o modo de controle do bloco podendo ser valor ou sensor, abrindo algumas opções para o controle.

2- Esta opção, quando desmarcada, faz com que os caminhos do bloco virem abas e estes fiquem mais compactos.

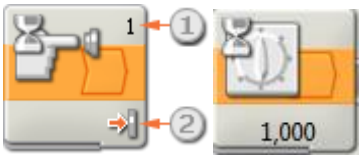
3- Esta opção é a configuração do seu controle. Como na imagem foi escolhido o tipo logic. vemos que este pode ser “verdadeiro” ou “falso”.

4- Esta opção escolhe o valor principal, aquele que será a condição para continuar no programa principal.

5.8 - WAIT BLOCK:

O Wait Block faz parte do conjunto de Flow Blocks, que são blocos que dinamizam as ações e o movimento do seu robô, tornando-o mais inteligente. Esse bloco permite que seu robô aguarde por um determinado tempo antes de tomar uma ação, ou que ele aguarde até que certa condição seja satisfeita, para que ele tome uma ação definida. Observe que os servo-motores não podem ser controlados diretamente pelo Wait Block, tendo que ser monitorados pela opção de tempo no Wait Block.

O Wait Block vem com duas formas de controle: Aguardar algum sensor (esquerda), e somente aguardar (direita).



Na função de aguardar algum sensor:

1. O número no canto superior direito mostra qual porta do NXT o Wait Block está monitorando. A porta pode ser mudada na opção **Port**.

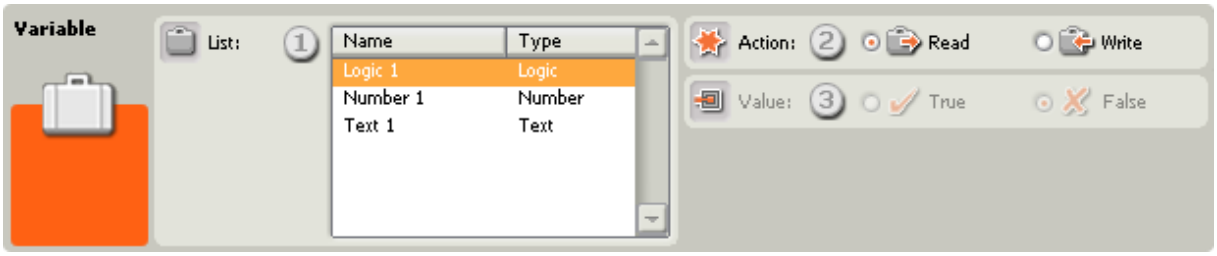
2. Mostra qual sensor que você está monitorando. Para cada sensor, existem configurações diferentes, mas que são bem parecidas com as configurações dos próprios blocos de sensores. No caso do Sensor de Toque, o ícone mostrará qual é a condição que está sendo monitorada (Botão apertado, pressionado ou encostado).

5.9 - VARIABLE BLOCK:

Uma variável é um espaço que você reserva na memória do seu computador para armazenar um determinado valor. Este é o princípio deste bloco, armazenar um determinado tipo de valor que você precisa.

1- O número indicado na figura representa a saída de dados do bloco, ou seja, você pode utilizar os dados do bloco usando um fio de dados e o ligando em outro bloco que possa utilizar esses dados.

Configurações do bloco



1- O número 1 da figura indica o tipo da variável que será usada.
2- Na opção action você pode escolher se a variável irá receber um valor ou se vai ser lido um valor armazenado nela.
3- Nesta opção você pode inserir diretamente um valor para a variável de acordo com o tipo.

5.10 - MATH BLOCK:

Este bloco executa operações aritméticas simples como adição, subtração, multiplicação e divisão. O bloco também pode executar valor absoluto e a raiz quadrada. Os números de entrada podem ser digitados ou fornecidos dinamicamente por fios de dados.

Na figura podemos observar duas entradas e uma saída, ou seja, nosso bloco permite a entrada de 2 números, dados, que serão fornecidos e de acordo com a escolha da operação irá gerar uma resultado de saída.

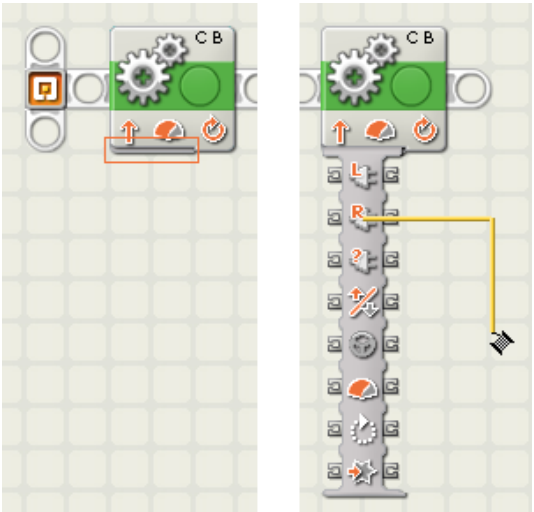
5.11- Number To Text BLOCK

Bloco usado para a conversão de números para caracteres; isso serve quando desejamos colocar no display do nosso Lego alguma informação numérica.

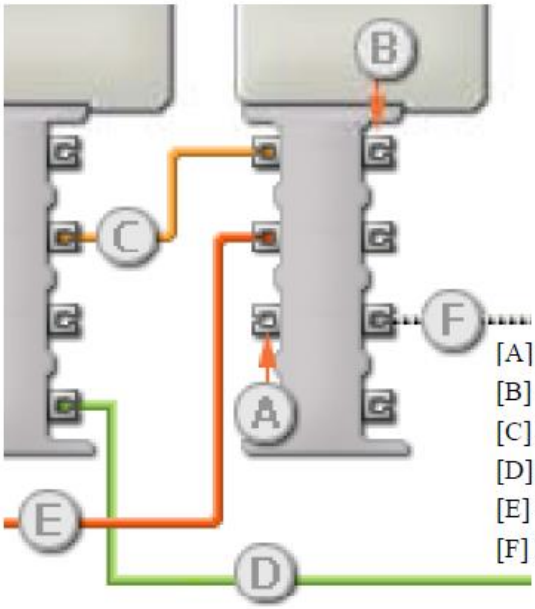
Os dados podem ser usados de blocos cujas saídas são números, e fazemos uma ligação por fios para a entrada #, ou podemos jogar um número diretamente na opção do bloco como na figura abaixo.

5.11 - USANDO FIOS DE DADOS (Data Wires)

Fios de dados transportam informações entre os blocos de programação. Algumas condições são necessárias para a utilização correta, o que veremos abaixo:



Podemos utilizar os fios de dados abrindo a aba de opções dos blocos para transportar diferentes informações. Cada tipo de dado tem uma respectiva entrada e saída, e isso varia conforme o bloco. Por exemplo, não se deve transportar um dado numérico para uma entrada lógica, quando isso é feito, o próprio programa acusa, na forma de um fio cinza pontilhado, e quando o fio é corretamente transportado, este possui uma coloração específica para o tipo de dado.



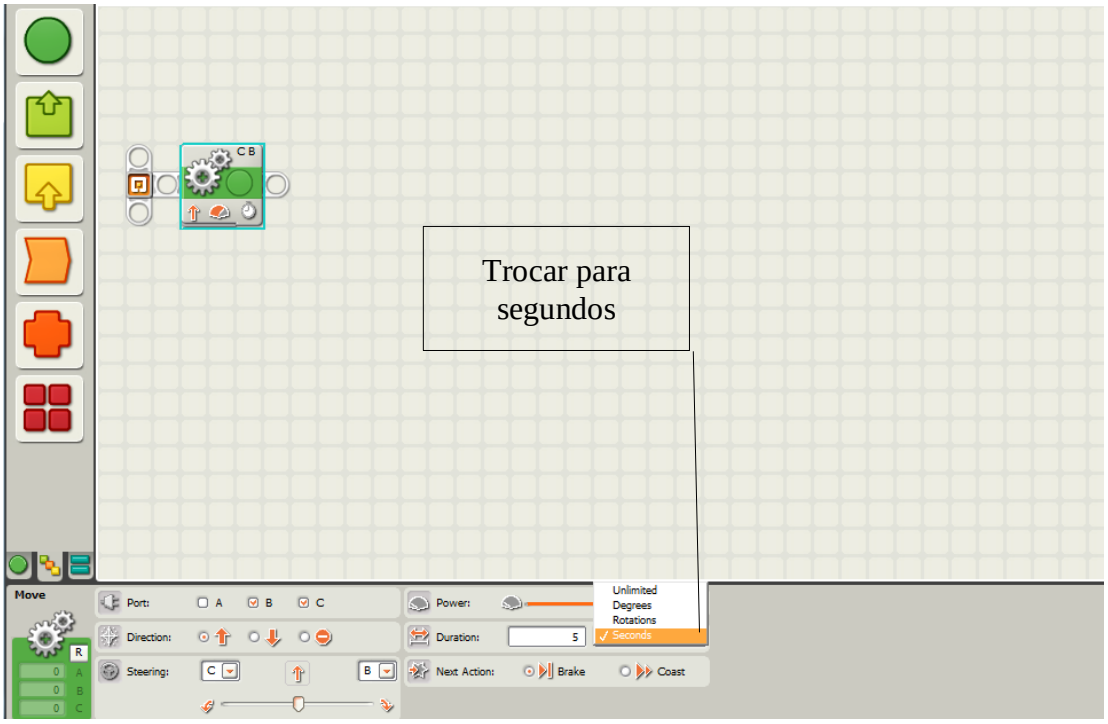
[A] Conexão de entrada (lado esquerdo do bloco)
[B] Conexão de saída (lado direito do bloco)
[C] Ligação de dados numéricos (amarelo)
[D] Ligação de dados lógicos (verde)
[E] Ligação de dados textos (laranja)
[F] Ligação de dados incompatíveis (erro-cinza)

EXEMPLOS DE PROGRAMAÇÃO

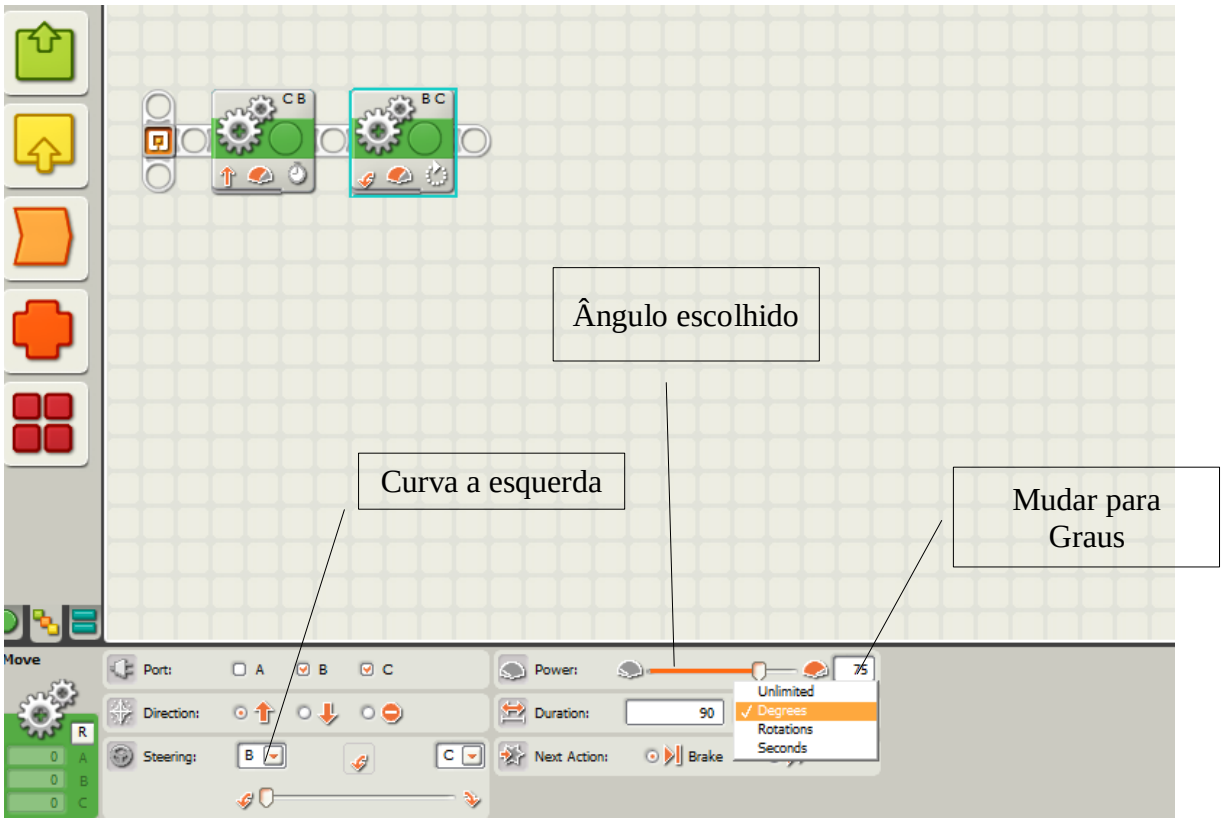
Exemplo 1 : Faça seu Robô andar durante 5 segundos com os motores B e C, depois gire 90° para a esquerda. Veja o passo a passo da montagem do robô "ExpressBot" no arquivo "Passo a passo - Exercicios de programação - ExpressBot.pdf".

Resolução.: Para resolvermos esse problema devemos separá-lo em partes afim de facilitar o processo;

- Vamos começar fazendo nosso robô andar durante 5 segundos com os motores B e C. Para isso podemos usar o bloco Move e configurá-lo para andar neste intervalo de tempo :



- Agora vamos utilizar o bloco Move novamente para fazer a curva para a esquerda:



- **Loops:**

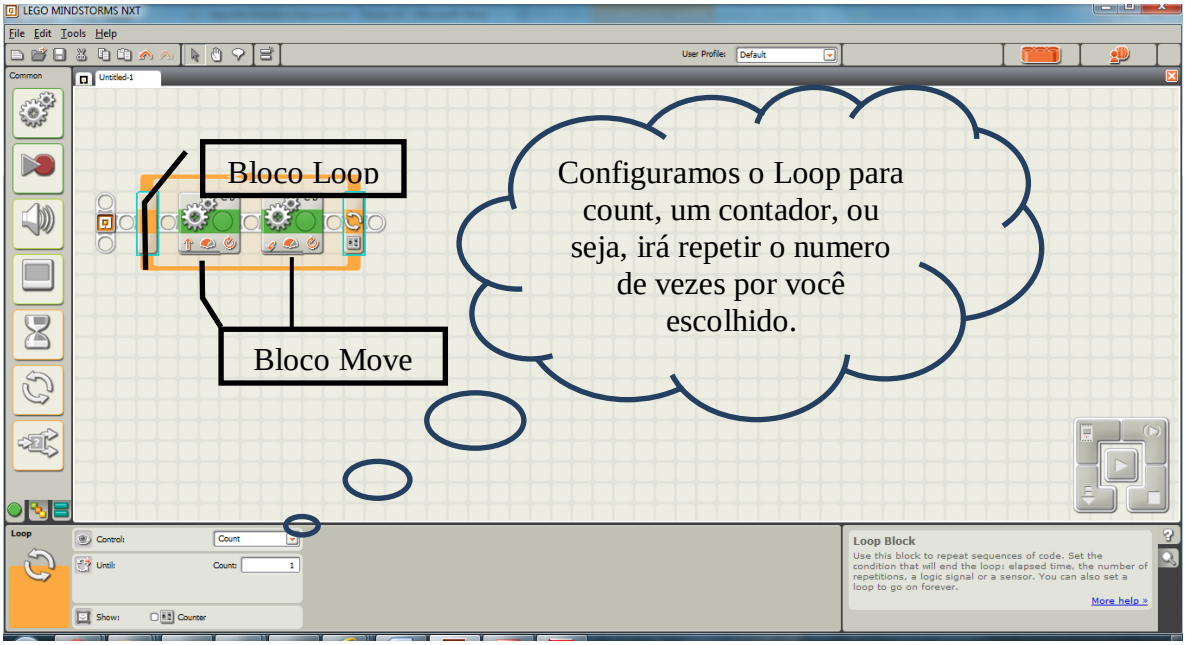
Imagine que você receba uma tarefa a ser cumprida, na qual você executa uma mesma atividade várias vezes como, por exemplo, pegar 20 canetas em seu armário e colocá-las em uma mesa na sala, sendo que só é permitido pegar uma por vez, o que você faria? Isso é simples, você iria andar até o armário pegar uma caneta e voltaria até a sala com a mesma colocando a caneta na mesa, e faria isso 20 vezes. Vale lembrar que nosso amigo robô não entende uma ordem deste tipo, ou seja, ele precisa receber essas 20 tarefas de forma independente. Mas, é claro, não é necessário que você repita a mesma tarefa 20 vezes.

Uma ferramenta muito usada por programadores em diversos ambientes de programação é o Loop, uma repetição, ou seja, essa ferramenta permite repetir uma atividade um determinado número de vezes imposto pelo programador; em nosso caso, nosso bloco muito usado em programas é o bloco Loop, que é configurado para fazer quantas repetições desejar.

Vamos exemplificar:

Exemplo – Robô: avance por 5 segundos, vire 90° para a esquerda. Repita quatro vezes.

Para este exemplo vamos usar três blocos, um será nosso bloco Loop e os outros para o movimento de nosso robô.

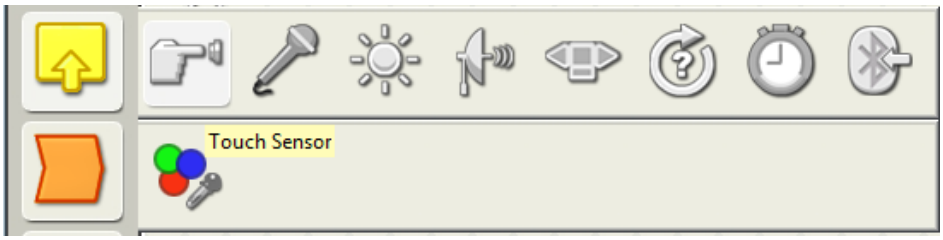


Configure o Loop para executar 4 vezes e os blocos de move de acordo com o que foi pedido.

Sensores

Com apenas estes blocos de movimento e repetição (Loop), já temos a possibilidade de criar vários programas. Porém, antes de nos aprofundarmos em problemas mais complexos, vamos voltar a comentar sobre os sensores, mas, desta vez, vamos utilizar seus blocos e ver alguns exemplos de seu funcionamento.

- **Sensor de Toque**

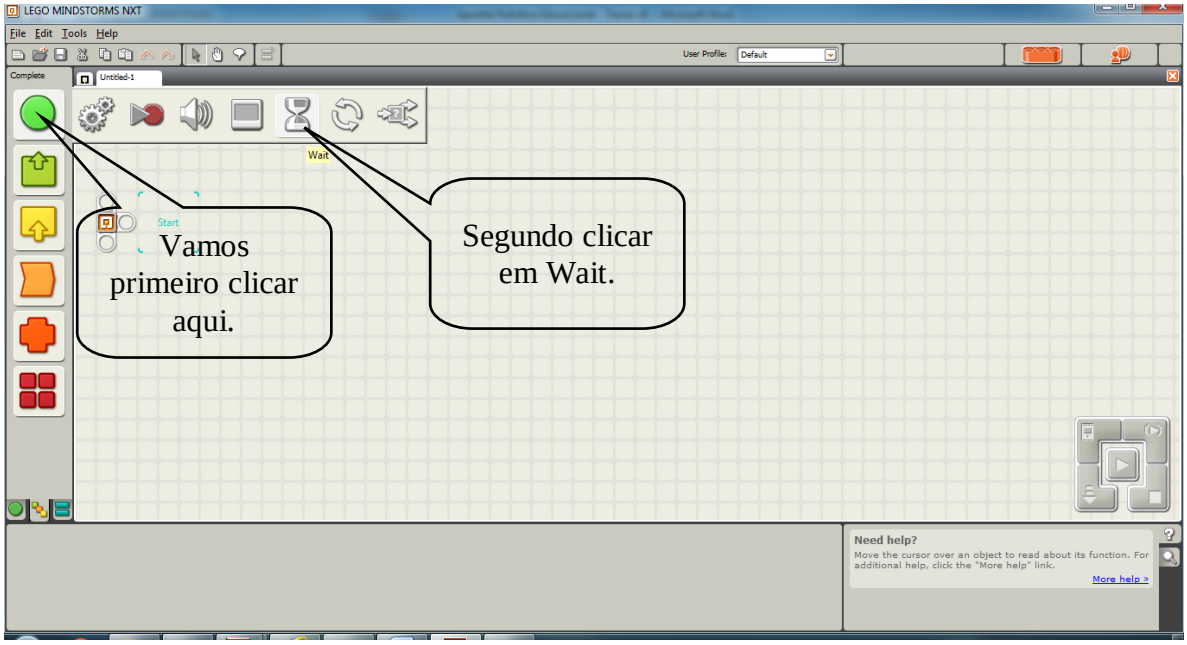


Vamos discutí-lo diretamente em um exemplo:

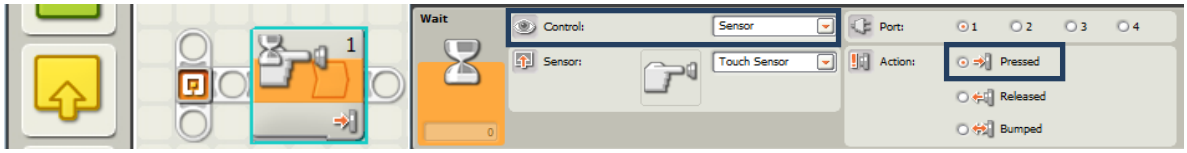
1° Exemplo: *Robô, ande até o sensor de toque a ser pressionado; quando esta condição for satisfeita, recuar 3 segundos e girar 180° para trás.*

Vamos apenas inserir o bloco que vai utilizar o sensor, o restante do programa fica como exercício.

Observamos que não se trata de um comando simples para colocar o bloco do sensor. Se prestarmos atenção, vamos entender que a condição é "andar até o botão ser pressionado", para isso vamos fazer o seguinte:



O bloco Wait é como um controlador, ele é muito usado para condicionar o movimento do seu robô. Escolhemos este em **control ,**marcamos **sensor** e em **action pressed.**



A ideia de usar os outros sensores para controlar seu programa é a mesma, você pode testar este mesmo programa com os outros sensores, adaptando cada situação com o sensor em questão. Basicamente, usaremos nossos sensores assim, como uma ferramenta que modificará o comportamento de nosso robô.

Existem muitos outros blocos dentro de nosso ambiente de trabalho no software, porém, nem todos esses serão vistos. Teremos uma ideia mais geral de outros blocos nos exercícios que estão por vir, que serão muito intuitivos e por isso evitaremos nos estender neste tópico.

Após abrir um novo arquivo, estamos prontos para começar a programar. Como exercício de programação, listamos os seguintes desafios abaixo. Não é necessária ainda nenhuma montagem com as peças do kit para realizar esses exercícios, apenas que os sensores estejam conectados ao NXT e que este esteja conectado ao computador.

EXERCÍCIOS DE PROGRAMAÇÂO:

1 - Usando o Display Block, faça com que o NXT exiba a mensagem “Hello world!’ em seu LCD por cinco segundos, depois espere um segundo e exiba um smile.

2 - Conectando um servo-motor ao NXT, faça com que o motor gire por cinco rotações para frente, aguarde dois segundos, gire de volta as cinco rotações e pare. Faça o NXT tocar um som avisando que completou o exercício.

6 - Experimento 1 - Práticas com o Sensor de Som NXT:

6.1 - Introdução:

Nesse primeiro experimento, é proposto que o grupo deverá se dividir entre construtores e programadores, para que a tarefa seja realizada mais rapidamente. É recomendável que os alunos troquem de funções a cada atividade, para ficarem aptos às duas etapas dos experimentos.

Neste experimento, será proposto que as equipes construam um modelo de veículo simples, para que possam controlá-lo através de som, usando o Sensor de Som. O passo a passo para a montagem é o arquivo "Passo a passo Experimento 1 – Base ModelBot.pdf". A programação será apresentada como pseudo-algoritmo, na seção "Entendendo o Programa", cabendo aos alunos a criação de seu próprio código em NXT-G. Para a montagem dos modelos, os alunos deverão acessar a seção *Robótica Educacional*, no site do *PET-Elétrica UFF* (www.peteletrica.uff.br), e baixar os arquivos em PDF.

6.2 - Sentidos e sensores:

Os cincos sentidos humanos:

Existem partes do seu corpo que atuam como órgãos sensoriais: olhos, nariz, ouvidos, língua e pele. Eles são responsáveis por levar informações ao seu cérebro sobre o ambiente ao seu redor, para que ele tome decisões. Isto é muito similar ao funcionamento dos sensores dos robôs. O seu cérebro usa continuamente as informações que recebe de seus órgãos sensoriais para fazer seu corpo funcionar corretamente.

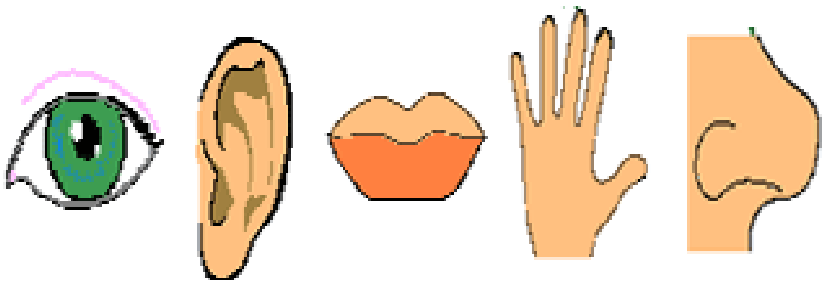


Figura 8.1 - Os cinco sentidos: visão, audição, paladar, tato e olfato

Quando os sensores do corpo humano detectam um estímulo, eles enviam essa informação através do sistema nervoso (semelhantes a fios) para o cérebro, que é o órgão que decide o que fazer com base nas informações recebidas.

Sensores Eletrônicos:

Os sensores dos robôs coletam informações dos arredores, através de pequenos componentes eletrônicos em seu interior, e enviam para o NXT Brick. Para enviarem essa informação, os sensores se comunicam através dos fios (semelhante ao do sistema nervoso no organismo humano) que os ligam ao NXT Brick, que utiliza a informação se o programa requer. Os Sensores também são muito parecidos com os membros do corpo humano (ex: Sensor ultrassônico e o sentido da visão; Sensor de toque e o sentido do tato, etc)

Figura 6.2 - NXT Brick e seus diversos sensores eletrônicos

Para o nosso experimento, iremos nos focar no uso do Sensor de Som, que é também o sentido auditivo do robô. Durante esta atividade, você vai ganhar mais experiência na programação dos seus robôs, desta vez para responder a diferentes sons!

Como funciona o Sensor de Som:

Ondas sonoras são vibrações mecânicas dissipadas no ar, que são captadas e interpretadas por nossos ouvidos. Toda onda sonora possui uma frequência própria, e quanto maior a frequência, mais agudo é o som. Quanto menor a frequência, mais graves são os sons.

No Sensor de Som, uma pequena camada fina de material elástico, chamada diafragma, vibra quando atingida por ondas sonoras (semelhante a um tímpano humano). Se

as vibrações no diafragma são grandes em volume o bastante para ser detectado, o sensor de som envia um sinal elétrico para o NXT Brick.

6.3 - Experimento proposto:

O experimento irá se focar no sentido da audição dos seres humanos e sua contraparte em um robô, o Sensor de Som. Usando o modelo "NXT Base ModelBot", os alunos são desafiados a criar um programa usando o sensor de som. O robô deve responder ao som de uma salva de palmas de mão. Ao ouvir a primeira palma, o robô, enquanto se move para frente, deve virar para a direita e, em seguida, continuar a avançar. Na segunda palma, deve virar à esquerda e continuar a avançar. Fazendo isso, eles melhoram seu entendimento do funcionamento do Sensor de Som, e reforçam seu conhecimento a respeito de processos sensoriais como um todo.

Figura 6.3 - Modelo NXT BaseModelbot com Sensor de som acoplado

6.4 - Entendendo o programa:

O programa necessário para o experimento deverá:

- ➔ **Movimentar-se** por tempo indeterminado;
- ➔ **Reconhecer** sons específicos e reagir a eles;
- ➔ **Parar** de se mexer quando pedido;
- ➔ **Girar** sobre o próprio eixo e voltar atrás;

Assim, o pseudo-algoritmo pode ser escrito da seguinte forma:

```
1    "Rodar o programa por quanto tempo quiser:
2    Se mover por quanto tempo quiser;
3    Se captar o som de palmas:
4        Dar meia-volta;
5        Se movimentar para o lado oposto até ouvir palmas novamente;
6    Repetir o passo 3;
```

6.5 - Atividades:

As atividades a serem realizadas estão descritas na *Folha de Atividades - Experimento 1 - Práticas com o Sensor de Som NXT*.

7 - Experimento 2 – Lançador NXT:

7.1 – Introdução:

Nesse experimento, será proposto que as equipes construam um disparador de projéteis simples que irá realizar lançamentos oblíquos, um experimento clássico de física cinemática. Com os lançamentos, os alunos poderão observar a eficiência dos servo-motores do NXT. O passo a passo para a montagem é o arquivo "Passo a passo Experimento 2 – NXT Ball Shooter.pdf", com algumas modificações que serão passadas pelos professores. A programação será apresentada como pseudo-algoritmo, na seção "Entendendo o Programa", cabendo aos alunos a criação de seu próprio código em NXT-G. Para a montagem dos modelos, os alunos deverão acessar a seção *Robótica Educacional*, no site do *PET-Elétrica UFF* (www.peteletrica.uff.br), e baixar os arquivos em PDF.

7.2 – Estudo do movimento dos corpos:

Um pouco de história:

As nossas ideias atuais sobre o movimento dos corpos vêm dos tempos de Galileu e de Newton. Além de provar, que, por exemplo, corpos de pesos diferentes caíam com velocidades diferentes (provando a existência de uma *força gravitacional g*, que se relaciona com a massa dos corpos), Galileu também é o responsável pelo **Princípio da independência dos movimentos**, ou Princípio de Galileu, que anuncia a independência dos movimentos na vertical e horizontal de um mesmo corpo e seus movimentos relativos, ou seja, esses dois movimentos podem ser estudados separadamente, pois eles não dependem um do outro.

Abaixo, um exemplo clássico do princípio: O lançamento oblíquo de um corpo.

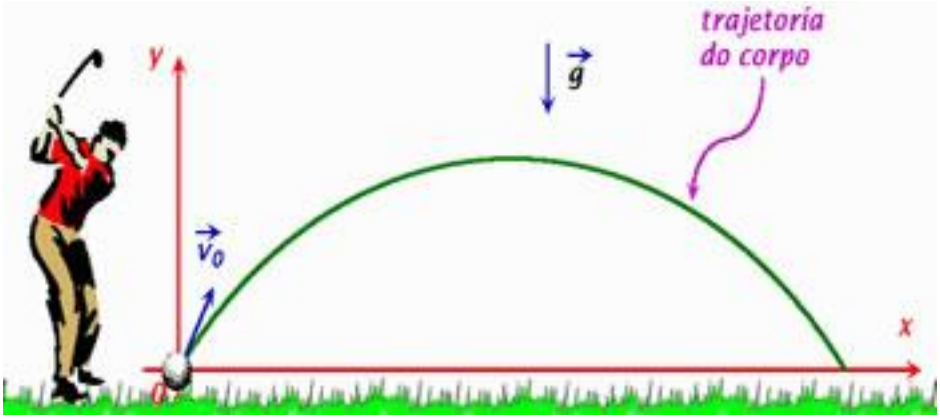


Figura 7.1 - Lancamento Oblíquo de um corpo

Observe que podemos separar o movimento oblíquo do corpo em dois: o movimento vertical e o movimento horizontal. O projétil se deslocará para a frente em uma trajetória que vai até uma altura máxima e depois volta a descer, formando uma trajetória parabólica..

Movimento horizontal:

Na direção horizontal o corpo realiza um Movimento Uniforme com velocidade igual a V_x. O alcance máximo é a distância entre o ponto do lançamento e o ponto da queda do corpo, e, por se tratar de um movimento uniforme, é determinado por: $A = V_x t$, sendo t a duração total do movimento.

Movimento vertical:

Na direção vertical o corpo realiza um Movimento Uniformemente Variado, com velocidade inicial igual a V_{oy}, e aceleração da gravidade (g). Durante a subida, a velocidade vertical diminui, chegando a um ponto de altura máxima H, onde $V_y=0$, e desce aumentando a velocidade, até chegar à mesma altura a qual foi lançado, onde volta a possuir a velocidade vertical inicial. Por ser um MUV, para determinarmos a altura máxima do movimento, utilizamos as equações de Função Horária:

$$H = V_{oy} t \pm \frac{g t^2}{2}$$

E a Equação de Torricelli:

$$V_y{}^2 = V_{oy}{}^2 \pm 2gH$$

Observe que a aceleração da gravidade será positiva ou negativa, dependendo se o projétil esta subindo (desacelerando), ou descendo (acelerando).

Para encontrarmos a velocidade instantânea em que o projétil foi disparado, faremos a soma vetorial das velocidades horizontal e vertical, ou seja, $V_o{}^2 = \sqrt{V_x{}^2 + V_{oy}{}^2}$. O vetor velocidade é tangente à trajetória em cada momento. Observe também que, pela equação acima, também podemos obter a seguinte relação trigonométrica entre as velocidades, sendo α o ângulo de lançamento do projétil:

$$V_o = V_x \cos \propto \quad \text{e} \quad V_o = V_{oy} \operatorname{sen} \propto$$

7.3 - Experimento Proposto:

O experimento desse capítulo irá propor a montagem de um lançador de projeteis, que atua em um ângulo determinado. Os alunos deverão fazer a montagem de acordo com o passo a passo, e tentar observar algumas grandezas relacionadas ao movimento do projétil. Após obtidas essas grandezas, os alunos irão alterar a potência oferecida aos servo-motores em seus programas, para que acertem alvos a distâncias menores ou maiores. Um gráfico relativo a potência do servo-motor e o alcance das bolinhas disparadas deverá ser traçado ao final.

Figura 7.2 - Modelo NXT Ball Shooter usado para lançamentos oblíquos

7.4 - Entendendo o programa:

O programa necessário para o experimento deverá:

- ➔ **Apresentar** instruções na tela de como atuar;
- ➔ **Responder** a algum comando para disparar (Ex: Botões NXT);
- ➔ **Mover** o servo-motor de forma precisa, disparando um projetil por vez;
- ➔ **Repetir** os disparos quantas vezes quiser;

Obs: Iniciaremos com os motores a 50% de potência.

7.5 - Atividades:

As atividades a serem realizadas estão descritas na *Folha de Atividades - Experimento 2 - Lançador de projeteis NXT.*

8 - Experimento 3 - Usando os Sensores Eletrônicos NXT

8.1 - Introdução:

Neste experimento, será proposto que as equipes construam um modelo de veículo, aonde seus movimentos são controlados pelos sensores ultrassônico e de toque. O passo a passo para a montagem é o arquivo "Tutorial Experimento 4 - Explorer.pdf". A programação será apresentada como pseudo-algoritmo, na seção "Entendendo o Programa", cabendo aos alunos a criação de seu próprio código em NXT-G. Para a montagem dos modelos, os alunos deverão acessar a seção *Aulas de Robótica Educacional*, no site do *PET-Elétrica UFF* (www.peteletrica.uff.br), e baixar os arquivos em PDF.

8.2 – O Sensor Ultrassônico

Vamos abordar o conceito físico de ondas, que como veremos posteriormente, será usado em nosso modelo escolhido.

Ondas se dividem em três tipos principais. Vamos abordar apenas dois tipos deles, pois o outro não é de nosso interesse neste material.

Ondas mecânicas: Essas ondas são as mais conhecidas por nós, já que estão presentes em grande parte de nosso dia-dia. São, por exemplo, as ondas do mar, as ondas sonoras e as sísmicas. Todas tem características em comum: obedecem as Leis de Newton e existem apenas em meios materiais, como a água, o ar e as rochas.

Ondas eletromagnéticas: Essas ondas podem ser menos conhecidas, mas são muito usadas. Podemos destacar a luz visível, as ondas de rádio e televisão, as micro-ondas, os raios X e as ondas de radar. Elas não precisam de um meio físico para se propagar; o maior exemplo disso é a luz solar que atravessa o vácuo e chega até nós.

Um sensor que utiliza o conceito de ondas em nosso experimento é o sensor ultrasônico, que gera ondas eletromagnéticas que se propagam até encontrarem algum obstáculo, sendo refletidas de volta e, com esses dados que chegam ao sensor, podemos realizar diversas aplicações.

Abaixo segue uma imagem que mostra basicamente o funcionamento do Explorer. Este utiliza o conceito de reflexão de uma onda, em que temos que nosso sensor, ao mesmo

tempo em que cria o pulso de ondas, também capta o mesmo, e utiliza-o no seu funcionamento.

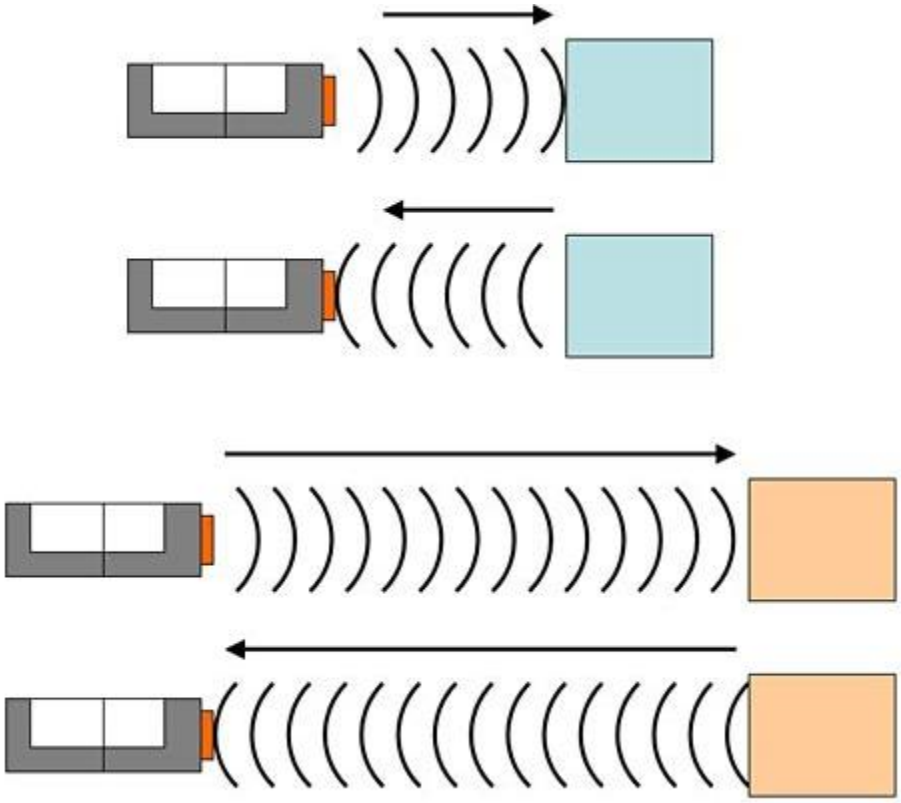


Figura 8.1 - Funcionamento do sensor

8.3 - Experimento proposto:

O experimento irá se focar ao uso do sensor ultrassônico e o Sensor de toque. Usando o modelo "ExplorerBot", os alunos são desafiados a criar um programa usando os sensores citados. O robô deve desviar de objetos a sua frente, de maneira que o sensor ultrassônico seja usado para visualizar objetos de certa forma altos, já o sensor de toque serve para os objetos muito baixos que ficam fora do alcance do ultrassônico.

Figura 8.2 - ExplorerBot

8.4 - Entendendo o programa

- ➔ **Movimentar-se** por tempo indeterminado;
- ➔ **Reconhecer** objetos específicos e reagir a eles;
- ➔ **Parar** de se mexer quando visualizado um objeto;
- ➔ **Olhar para os lados** escolher o lado mais limpo de obstáculos e ir nesta direção;

Assim, o pseudo-algoritmo pode ser escrito da seguinte forma:

```
1     "Rodar o programa por quanto tempo quiser:
2     Se mover por quanto tempo quiser;
3     Se captar um objeto com o sensor ultrassônico:
4           parar, olhar para os lados;
5           seguir na direção com menos obstáculos;
6     Se o sensor de toque dor ativado:
7           dar ré;
8           olhar para os lados seguir na direção com menos obstáculos
9     Repetir o passo 3;
```

9 - Experimento 4 - Fita Métrica Eletrônica NXT:

9.1 - Introdução:

Nesse experimento, os alunos irão construir um equipamento simples, capaz de medir o comprimento de superfícies retangulares, medindo em duas ou três dimensões. Usando essas medidas, o NXT sozinho irá calcular a área e volume dessas superfícies, e apresentar na tela de LCD o resultado. O passo a passo para a montagem é o arquivo "Passo a passo Experimento 4 – NXT Area and Volume Calculator.pdf". A programação será apresentada como pseudo-algoritmo, na seção "Entendendo o Programa", cabendo aos alunos a criação de seu próprio código em NXT-G. Para a montagem dos modelos, os alunos deverão acessar a seção *Robótica Educacional*, no site do *PET-Elétrica UFF* (www.peteletrica.uff.br), e baixar os arquivos em PDF.

9.2 - Curiosidades sobre medidas e unidades:

O corpo humano como unidade:

As primeiras unidades de medida referiam-se direta ou indiretamente ao corpo humano. Por volta de 3500 a.C. - quando na Mesopotâmia e no Egito começaram a ser construídos os primeiros templos - seus projetistas tiveram de encontrar unidades mais uniformes e precisas. Adotaram a longitude das partes do corpo de um único homem (geralmente o rei) e com essas medidas construíram réguas de madeira e metal, ou cordas com nós, que foram as primeiras medidas oficiais de comprimento.

São exemplos de unidades de medidas antigas, e hoje pouco utilizadas:

Jarda (distância do centro do peito até a ponta dos dedos): **1 jarda = 91,44 cm;**

Polegada (usada ainda no sistema americano de medidas): **1 polegada = 30,48 cm;**

Pé (usada ainda no sistema americano de medidas): **1 pé = 2,54 cm;**

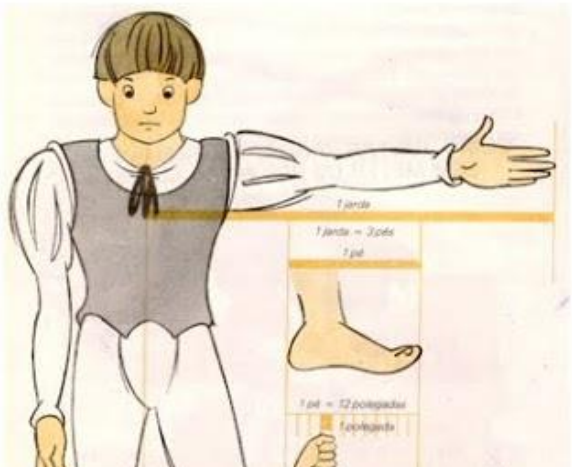


Figura 9.1 - Unidades de medidas antigas

Para medir superfícies:

Os sacerdotes dos templos daquele tempo, encarregados de arrecadar os impostos sobre a terra, provavelmente começaram a calcular a extensão dos campos por meio de um simples golpe de vista. Certo dia, ao observar trabalhadores pavimentando uma superfície retangular com mosaicos quadrados, algum sacerdote deve ter notado que, para conhecer o número total de mosaicos, bastava contar os de uma fileira e repetir esse número tantas vezes quantas fileiras houvesse. Assim nasceu a fórmula da área do retângulo: multiplicar a base pela altura.

Já para descobrir a área do **triângulo**, os antigos fiscais seguiram um raciocínio extremamente geométrico. Para acompanhá-lo, basta tomar um quadrado ou um retângulo e dividí-lo em quadradinhos iguais. Suponhamos que o quadrado tenha 9 "casas" e o retângulo 12. Esses números exprimem então a área dessas figuras. Cortando o quadrado em duas partes iguais, segundo a linha diagonal, aparecem dois triângulos iguais, cuja área, naturalmente, é a metade da área do quadrado.

E claro, haviam construções que requeriam uma parede curva. Assim, um novo problema se apresenta: como determinar o comprimento de uma circunferência e a área de um círculo. Por circunferência entende-se a linha da periferia do círculo, sendo este uma superfície. Já os antigos geômetras observavam que, para demarcar círculos, grandes ou pequenos, era necessário usar uma corda, longa ou curta, e girá-la em torno de um ponto fixo,

que era a estaca cravada no solo como centro da figura. O comprimento dessa corda - conhecido hoje como **raio** - tinha algo a ver com o comprimento da circunferência. Retirando a corda da estaca e colocando-a sobre a circunferência para ver quantas vezes cabia nela, puderam comprovar que cabia um pouco mais de seis vezes e um quarto (6,28). Qualquer que fosse o tamanho da corda, o resultado era o mesmo.

E a **área do círculo**? A história da Geometria explica-a de modo simples e interessante. Cerca de 2000 anos a.C., um escrivão egípcio chamado Ahmes raciocinava diante do desenho de um círculo no qual havia traçado o respectivo raio. Seu propósito era encontrar a área da figura.

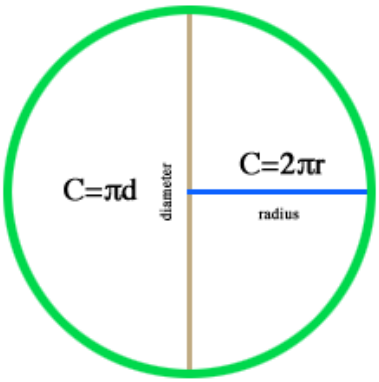


Figura 9.2 - Representação de raio e diâmetro

Conta a tradição que Ahmes solucionou o problema facilmente: antes, pensou em determinar a área de um quadrado e calcular quantas vezes essa área caberia na área do círculo. Que quadrado escolher? Qualquer um? Parecia razoável tomar o que tivesse como lado o próprio raio da figura. Assim fez, e comprovou que o quadrado estava contido no círculo mais de 3 vezes e menos de 4, ou aproximadamente, três vezes e um sétimo (atualmente dizemos 3,14 vezes). Concluiu então que, para saber a área de um círculo, basta calcular a área de um quadrado construído sobre o raio (raio do quadrado), e multiplicar a respectiva área por 3,14.

O **número 3,14** é básico na Geometria e na Matemática. Os gregos tornaram-no um pouco menos inexato: 3,1416. Hoje, o símbolo π ("pi") representa esse número irracional, já determinado com uma aproximação de várias dezenas de casas decimais. Seu nome só tem cerca de duzentos anos e foi tirado da primeira sílaba da palavra grega ***peripheria*,** significando circunferência.

9.3 - Experimento proposto:

O modelo "NXT Area and Volume Calculator" consiste de uma roda anexada ao NXT, que mede a distância percorrida pela roda. Construa uma fita métrica inteligente, que mede comprimento de superfícies, e que calcula sozinha a área entre essas grandezas e o volume, além de apresentar os resultados em tempo real no display. Quantos metros cúbicos você acha que possui a sala em que você se encontra? Tente medir isso!

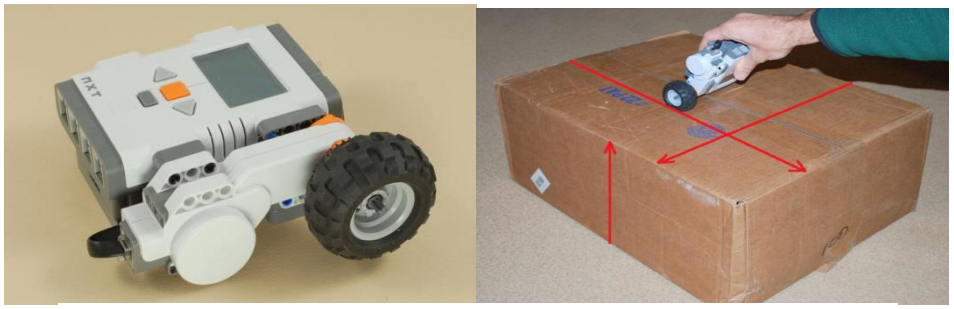


Figura 6.3 - Modelo NXT Area and Volume Calculator

9.4 - Entendendo o programa:

O programa necessário para a montagem deverá:

- ➔ **Realizar** as medidas quantas vezes quiser;
- ➔ **Relacionar** o giro da engrenagem do servo-motor NXT ao giro da roda de medição (lembrando que a circunferência da roda é dada por $C = 2\pi R$);
- ➔ **Armazenar** o valor medido para cada uma das grandezas;
- ➔ **Entender** quando que uma medição foi completada (para que possa fazer a próxima);
- ➔ **Multiplicar** esses valores para obter Área e Volume;
- ➔ **Exibir** os valores que estão sendo medidos (e o resultado de área e volume), **enquanto** eles estão sendo medidos.

Observe que, para que o NXT entenda que cada medição foi completada, ele precisa ser *condicionado a isso*, ou seja, uma condição proposta pelo programador deve ser cumprida, para que ele siga adiante com o programa. Caso contrário, o NXT nunca iria determinar que uma medição fosse terminada, mesmo que o usuário termine a medição por conta própria.

Assim, o pseudo-algoritmo pode ser escrito da seguinte forma:

1 "Fazer as medidas quantas vezes quiser:

2 Apresentar para o usuário qual botão ele deve apertar para realizar as medidas;

3 Resetar o giro do Sensor de Rotação do Servo-motor;

4 Enquanto não pressionar o botão "Próximo":

5 Medir a rotação do motor com o Sensor de Rotação;

6 Calcular a rotação da roda;

7 Guardar esse valor;

8 Transformar esse valor obtido (Número) em um valor Texto, que possa ser apresentado na tela;

9 Escrever e apresentar na tela esse valor, especificando o que ele é (comprimento, altura, largura, área ou volume);

10 Após o usuário pressionar o botão "próximo", refazer os passos 3 (que deve ser feito separado),e de 5 a 9, e depois:

11 Calcular outro valor (área), com os dois valores já guardados;

12 Repetir passos 7 a 9;

13 Repetir passo 10;

14 Repetir passo 11, mas agora, usando os dois últimos valores que foram guardados (área + comprimento/largura/altura), calcular um terceiro valor (volume);

15 Repetir passos 8 e 9".

9.5 - Atividades:

As atividades a serem realizadas estão descritas na *Folha de Atividades - Experimento 4 - Fita Métrica Eletrônica NXT.*

10 - Desafios

Nesta seção vamos pôr em prática todo o conhecimento em relação à parte de montagem e programação em LEGO Mindstorms NXT adquiridos até aqui. Seguem abaixo alguns desafios que as equipes de estudantes deverão executar. A montagem será fornecida pelos professores, mas a programação será totalmente original. Usem sua criatividade para resolver os problemas apresentados!